# Chapter 5
# Luminance: a New Visual Feature for Visual Servoing

Christophe Collewet and Eric Marchand

**Abstract** This chapter is dedicated to a new way to achieve robotic tasks by 2D visual servoing. Contrary to most of related works in this domain where geometric visual features are usually used, we directly here consider the luminance of all pixels in the image. We call this new visual servoing scheme *photometric visual servoing*. The main advantage of this new approach is that it greatly simplifies the image processing required to track geometric visual features all along the camera motion or to match the initial visual features with the desired ones. However, as it is required in classical visual servoing, the computation of the so-called *interaction matrix* is required. In our case, this matrix links the time variation of the luminance to the camera motions. We will see that this computation is based on a illumination model able to describe complex luminance changes. However, since most of the classical control laws fail when considering the luminance as a visual feature, we turn the visual servoing problem into an optimization one leading to a new control law. Experimental results on positioning tasks validate the feasibility of photometric visual servoing and show its robustness regarding to approximated depths, Lambertian and non Lambertian objects, low textured objects, partial occlusions and even, to some extent, to images content.

## 5.1 Introduction

Visual servoing is now a widely used technique is robot control [4]. More generally, it consists in using information provided by a vision sensor to control the state of a dynamic system. Robust extraction and real-time spatio-temporal tracking of visual cues is then usually one of the keys to success of a visual servoing task. We will show here that this tracking process can be totally removed and that no

Christophe Collewet and Eric Marchand
INRIA, Campus de Beaulieu, 35042 Rennes, France, e-mail: {christophe.collewet,eric.marchand}@inria.fr

other information than the image intensity (that is the pure luminance signal) can be considered to control the robot motion.

Classically, to achieve a visual servoing task, a set of visual features has to be selected from the image in order to control the desired degrees of freedom (DOF). A control law has also to be designed so that these visual features $\mathbf{s}$ reach a desired value $\mathbf{s}^*$, leading to a correct realization of the task. The control principle is thus to regulate to zero the error vector $\mathbf{e} = \mathbf{s} - \mathbf{s}^*$. To build this control law, the interaction matrix $\mathbf{L_s}$ is required. For eye-in-hand systems, this matrix links the time variation of $\mathbf{s}$ to the camera instantaneous velocity $\mathbf{v}$

$$\dot{\mathbf{s}} = \mathbf{L_s}\,\mathbf{v} \tag{5.1}$$

with $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ where $\boldsymbol{v}$ is the linear camera velocity and $\boldsymbol{\omega}$ its angular velocity. Thereafter, if we consider the camera velocity as input of the robot controller, the following control law is designed to try to obtain an exponential decoupled decrease of the error $\mathbf{e}$

$$\mathbf{v} = -\lambda \widehat{\mathbf{L}_\mathbf{s}^+}\,\mathbf{e} \tag{5.2}$$

where $\lambda$ is a proportional gain that has to be tuned to minimize the time-to-convergence, and $\widehat{\mathbf{L}_\mathbf{s}^+}$ is the pseudo-inverse of a model or an approximation of $\mathbf{L_s}$ [4].

As it can be seen, visual servoing explicitly relies on the choice of the visual features $\mathbf{s}$ (and then on the related interaction matrix); that is the key point of this approach. However, with a vision sensor providing 2D measurements $\mathbf{x}(\mathbf{r}_k)$ (where $\mathbf{r}_k$ is the camera pose at time $k$), potential visual features $\mathbf{s}$ are numerous, since 2D data (coordinates of feature points in the image, contours, moments,...) as well as 3D data provided by a localization algorithm exploiting $\mathbf{x}(\mathbf{r}_k)$ can be considered. In all cases, if the choice of $\mathbf{s}$ is important, it is always designed from the visual measurements $\mathbf{x}(\mathbf{r}_k)$. However, a robust extraction, matching (between $\mathbf{x}(\mathbf{r}_0)$ and $\mathbf{x}^* = \mathbf{x}(\mathbf{r}^*)$ where $\mathbf{r}^*$ is the camera desired pose) and real-time spatio-temporal tracking (between $\mathbf{x}(\mathbf{r}_{k-1})$ and $\mathbf{x}(\mathbf{r}_k)$) have proved to be a complex task, as testified by the abundant literature on the subject (see [17] for a recent survey on this subject). This image processing is, to date, a necessary step and considered also as one of the bottlenecks of the expansion of visual servoing. That is why some works tend to alleviate this problem. A first idea is to select visual features as proposed in [11, 14] or as in [19] to only keep visual features that are tracked with a high confident level (see also [7] where a more general approach is proposed). However, the goal of such approaches is not to simplify the image processing step but to take into account that it can fail. A more interesting way to avoid any tracking process is to use non geometric visual features. In that case, parameters of a 2D motion model are used as in [21, 24, 23, 8]. Nevertheless, such approaches require an important and complex image processing step. Removing the entire matching process is only possible when using directly the luminance as we propose.

Indeed, to achieve this goal we use as visual features the simplest feature that can be considered: the image intensity itself. We therefore called this new approach *photometric visual servoing*. In that case, the visual feature vector $\mathbf{s}$ is nothing but

the image while $\mathbf{s}^*$ is the desired image. The error $\mathbf{e}$ is then only the difference between the current and desired images (that is $\mathbf{e} = \mathbf{I} - \mathbf{I}^*$ where $\mathbf{I}$ is a vector that contains image intensity of all pixels).

However, considering the whole image as a feature has previously been considered [18, 9]. As in our case, the methods presented in [9, 18] did not require a matching process. Nevertheless they differ from our approach in two important points. First, they do not use directly the image intensity since an eigenspace decomposition is performed to reduce the dimensionality of image data. The control is then performed in the eigenspace and not directly with the image intensity. Moreover, this way to proceed requires the off-line computation of this eigenspace and then, for each new frame, the projection of the image on this subspace. Second, the interaction matrix related to the eigenspace is not computed analytically but learned during an off-line step. This learning process has two drawbacks: it has to be done for each new object and requires the acquisition of many images of the scene at various camera positions. Considering an analytical interaction matrix avoids these issues.

An interesting approach, which also consider the pixels intensity, has been recently proposed in [15]. This approach is based on the use of kernel methods that lead to a high decoupled control law. However, only the translations and the rotation around the optical axis are considered whereas, in our work, the 6 DOF are controlled. Another approach that does not require tracking nor matching has been proposed in [1]. It models collectively feature points extracted from the image as a mixture of Gaussian and try to minimize the distance function between the Gaussian mixture at current and desired positions. Simulation results show that this approach is able to control the 3 DOF of robot (and the 6 DOF under some assumptions). However, note that an image processing step is still required to extract the current feature points. Our approach does not require this step. Finally, in [2], the authors present an homography-based approach to visual servoing. In this method the image intensity of a planar patch is first used to estimate the homography between current and desired image which is then used to build the control law. Despite the fact that, as in our case, image intensity is used as the basis of the approach, an important image processing step is necessary to estimate the homography. Furthermore, the visual features used in the control law rely on the homography matrix and not directly on the luminance.

In the remainder of this chapter we first compute the interaction matrix related to the luminance in Section 5.2. Then, we reformulate the visual servoing problem into an optimization problem in Section 5.3 and propose a new control law dedicated to the specific case of the luminance. Section 5.4 shows experimental results on various scenes for several positioning tasks.

## 5.2 Luminance as a Visual Feature

The visual features that we consider here are the luminance $I$ of each point of the image, that is

$$\mathbf{s}(\mathbf{r}) = \mathbf{I}(\mathbf{r}) = (\mathbf{I}_{1\bullet}, \mathbf{I}_{2\bullet}, \cdots, \mathbf{I}_{N\bullet}) \tag{5.3}$$

where $\mathbf{I}_{k\bullet}$ is nothing but the $k$-th line of the image. $\mathbf{I}(\mathbf{r})$ is then a vector of size $N \times M$ where $N \times M$ is the size of the image. As mentioned in Section 5.1, an estimation of the interaction matrix is at the center of the development of any visual servoing scheme. In our case, we have to derive the interaction matrix related to the luminance of a pixel in the image, that is

$$\lim_{dt \to 0} \frac{I(\mathbf{x}, t+dt) - I(\mathbf{x}, t)}{dt} = \mathbf{L}_I(\mathbf{x})\mathbf{v} \tag{5.4}$$

$\mathbf{x} = (x, y)$ being the normalized coordinates of the projection $\mathbf{p}$ of a point physical $\mathbf{P}$ belonging to the scene.

Before computing the interaction matrix $\mathbf{L}_I(\mathbf{x})$ in the general case, lets first consider the simpler case where the temporal luminance constancy hypothesis is assumed, as it is done in most of computer vision applications. Let us also assume that $\mathbf{p}$ has a small displacement $\mathbf{dx}$ in the time interval $dt$

$$I(\mathbf{x} + \mathbf{dx}, t + dt) = I(\mathbf{x}, t). \tag{5.5}$$

If $\mathbf{dx}$ is small enough, a first order Taylor series expansion of (5.5) around $\mathbf{x}$ can be performed yielding the so-called *optical flow constraint equation* (OFCE) [13]

$$\nabla I^\top \dot{\mathbf{x}} + I_t = 0 \tag{5.6}$$

with $\nabla I$ the spatial gradient of $I(\mathbf{x}, t)$[1] and $I_t = \partial I(\mathbf{x}, t)/\partial t$. Moreover, considering the interaction matrix $\mathbf{L}_\mathbf{x}$ related to $\mathbf{x}$ (i.e. $\dot{\mathbf{x}} = \mathbf{L}_\mathbf{x}\mathbf{v}$)

$$\mathbf{L}_\mathbf{x} = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \tag{5.7}$$

(5.6) gives

$$I_t = -\nabla I^\top \mathbf{L}_\mathbf{x}\mathbf{v}. \tag{5.8}$$

However, note that $I_t$ is nothing but the left part of (5.4). Consequently, from (5.4) and (5.8), we obtain the interaction matrix $\mathbf{L}_I(\mathbf{x})$ related to $I$ at pixel $\mathbf{x}$

$$\mathbf{L}_I(\mathbf{x}) = -\nabla I^\top \mathbf{L}_\mathbf{x}. \tag{5.9}$$

Of course, because of the hypothesis required to derive (5.5), (5.9) can only be valid for Lambertian scenes, that is for surfaces reflecting the light with the same

[1] Let us point out that the computation of $\nabla I$ is the *only* image processing step necessary to implement our method.

intensity in each direction. Besides, (5.9) is also only valid for a motionless lighting source with respect to the scene.

Indeed, to derive the interaction matrix in the general case, we have to consider a more realistic reflection model than the Lambert's one. The Lambert's model can only explained the behavior of non homogeneous opaque dielectric material [22]. It only describes a diffuse reflection component and does not take into account the viewing direction. We propose to use the well-known Phong model [20]. However, note that this model is not based on physical laws, but comes from the computer graphics community. Although empirical, it is widely used thanks to its simplicity, and because it is appropriate for various types of materials, whether they are rough or smooth. Note that other models could be considered such as the Blinn-Phong [3] as reported in [5].

According to the Phong model (see Fig. 5.1), the intensity $I(\mathbf{x})$ at point $\mathbf{x}$ writes as follows

$$I(\mathbf{x}) = K_s \cos^k \alpha + K_d \cos\theta + K_a. \tag{5.10}$$

This relation is composed of a diffuse, a specular and an ambient component and assumes a point light source. The scalar $K_s$ describes the specular component of the lighting; $K_d$ describes the part of the diffuse term which depends on the *albedo* in $\mathbf{P}$; $K_a$ is the intensity of ambient lighting in $\mathbf{P}$. Note that $K_s, K_d$ and $K_a$ depend on $\mathbf{P}$. $\theta$ is the angle between the normal to the surface $\mathbf{n}$ in $\mathbf{P}$ and the direction of the light source $\mathbf{L}$; $\alpha$ is the angle between $\mathbf{R}$ (which is $\mathbf{L}$ mirrored about $\mathbf{n}$) and the viewing direction $\mathbf{V}$. $\mathbf{R}$ can be seen as the direction due to a pure specular object, where $k$ allows to model the width of the specular lobe around $\mathbf{R}$, this scalar varies as the inverse of the roughness of the material.

In the remainder of this chapter, the unit vectors $\mathbf{i}, \mathbf{j}$ and $\mathbf{k}$ correspond to the axis of the camera frame (see Fig. 5.1).

Considering that $\mathbf{R}, \mathbf{V}$ and $\mathbf{L}$ are normalized, we can rewrite (5.10) as

$$I(\mathbf{x}) = K_s {u_1}^k + K_d u_2 + K_a \tag{5.11}$$

where $u_1 = \mathbf{R}^\top \mathbf{V}$ and $u_2 = \mathbf{n}^\top \mathbf{L}$. Note that these vectors are easy to compute, since we have

$$\mathbf{V} = -\frac{\tilde{\mathbf{x}}}{\| \tilde{\mathbf{x}} \|} \tag{5.12}$$

$$\mathbf{R} = 2u_2\mathbf{n} - \mathbf{L}. \tag{5.13}$$

with $\tilde{\mathbf{x}} = (x, y, 1)$.

In the general case, we consider the following dependencies

$$\begin{cases} \mathbf{V} = \mathbf{V}(\mathbf{x}(t)) \\ \mathbf{n} = \mathbf{n}(\mathbf{x}(t), t) \\ \mathbf{L} = \mathbf{L}(\mathbf{x}(t), t) \\ \mathbf{R} = \mathbf{R}(\mathbf{x}(t), t). \end{cases} \tag{5.14}$$

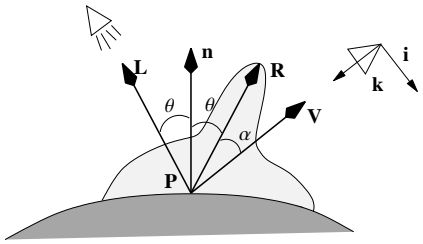


**Fig. 5.1** The Phong illumination model [20].

From the definition of the interaction matrix given in (5.4), its computation requires to write the total derivative of (5.11)

$$\dot{I} = kK_s u_1^{k-1} \dot{u}_1 + K_d \dot{u}_2. \tag{5.15}$$

However, it is also possible to compute $\dot{I}$ as

$$\dot{I} = \nabla I^\top \dot{\mathbf{x}} + I_t = \nabla I^\top \mathbf{L}_\mathbf{x} \mathbf{v} + I_t \tag{5.16}$$

where we have introduced the interaction matrix $\mathbf{L}_\mathbf{x}$ associated to $\mathbf{x}$. Consequently, from (5.15) and (5.16), we obtain

$$\nabla I^\top \mathbf{L}_\mathbf{x} \mathbf{v} + I_t = kK_s u_1^{k-1} \dot{u}_1 + K_d \dot{u}_2 \tag{5.17}$$

that is a general formulation of the OFCE considering the Phong illumination model.

Thereafter, by explicitly computing the total time derivative of $u_1$ and $u_2$ and writing

$$\dot{u}_1 = \mathbf{L}_1^\top \mathbf{v} \text{ and } \dot{u}_2 = \mathbf{L}_2^\top \mathbf{v}, \tag{5.18}$$

we obtain the interaction matrix related to the intensity at pixel $\mathbf{x}$ in the general case

$$\mathbf{L}_I = -\nabla I^\top \mathbf{L}_\mathbf{x} + kK_s {u_1}^{k-1} \mathbf{L}_1^\top + K_d \mathbf{L}_2^\top. \tag{5.19}$$

Note that we recover the interaction matrix $-\nabla I^\top \mathbf{L}_\mathbf{x}$ associated to the intensity under temporal constancy (see (5.9)), i.e. in the Lambertian case ($K_s = 0$) and when $\dot{u}_2 = 0$ (i.e. the lighting direction is motionless with respect to the point $\mathbf{P}$).

To compute the vectors $\mathbf{L}_1$ and $\mathbf{L}_2$ involved in (5.19) we have to explicitly express $\dot{u}_1$ and $\dot{u}_2$. However, to do that, we have to assume some hypothesis about how $\mathbf{n}$ and $\mathbf{L}$ move with respect to the observer. Various cases have been studied in [6]. Nevertheless, to make this chapter more readable, we report here only the case where the light source is mounted on the camera and only give the final equation. However, all the details can be found in [6].

In this case, we simply have $\mathbf{L} = -\mathbf{k}$ (see Fig. 5.1). After tedious computations, it can be shown that

$$\mathbf{L}_2^\top = -\nabla n_z^\top \mathbf{L}_\mathbf{x} + \mathbf{L}_4^\top \tag{5.20}$$

where $\mathbf{n} = (n_x, n_y, n_z)$ and $\mathbf{L}_4^\top = \begin{pmatrix} 0 & 0 & 0 & (\mathbf{n}\times\mathbf{k})^\top\mathbf{i} & (\mathbf{n}\times\mathbf{k})^\top\mathbf{j} & 0 \end{pmatrix}$. $\mathbf{L}_1^\top$ is expressed as follows:

$$\mathbf{L}_1^\top = (\mathbf{V}^\top\mathbf{J}^{\mathbf{R}} + \mathbf{R}^\top\mathbf{J}^{\mathbf{V}})\mathbf{L}_{\mathbf{x}} + \mathbf{L}_3^\top \tag{5.21}$$

where $\mathbf{J}^{\mathbf{R}}$ and $\mathbf{J}^{\mathbf{V}}$ are respectively the Jacobian matrices related to $\mathbf{R}$ and $\mathbf{V}$ (see [6]) with respect to $\mathbf{x}$, while $\mathbf{L}_3^\top = \begin{pmatrix} 0 & 0 & 0 & \mathbf{L}_{3x} & \mathbf{L}_{3y} & \mathbf{L}_{3z} \end{pmatrix}$ with

$$\begin{cases} \mathbf{L}_{3x} = 2(\mathbf{n}^\top\mathbf{V}(\mathbf{n}\times\mathbf{k})^\top + \mathbf{k}^\top\mathbf{n}(\mathbf{n}\times\mathbf{V})^\top)\mathbf{i} \\ \mathbf{L}_{3y} = 2(\mathbf{n}^\top\mathbf{V}(\mathbf{n}\times\mathbf{k})^\top + \mathbf{k}^\top\mathbf{n}(\mathbf{n}\times\mathbf{V})^\top)\mathbf{j} \\ \mathbf{L}_{3z} = 2\mathbf{k}^\top\mathbf{n}(\mathbf{n}\times\mathbf{V})^\top\mathbf{k}. \end{cases} \tag{5.22}$$

However, the interaction matrix is very often computed at the desired position [4]. Indeed, this way to proceed avoid to compute on-line 3D information like the depths for example. We also here consider this case. More precisely, we consider that, at the desired position the depth of all the points where the luminance is measured are equal to a constant value $Z^*$. That means that we consider that the object is planar and that the camera and the object planes are parallel at this position. This case is depicted on the Fig. 5.2. Here, since we suppose that $\mathbf{J}^{\mathbf{n}} = \mathbf{0}$ and $\mathbf{n} = -\mathbf{k}$, it is straightforward to show that $\mathbf{L}_2^\top = \mathbf{0}$. Besides, since $\mathbf{n} = -\mathbf{k}$ and $\mathbf{L} = -\mathbf{k}$, we have $\mathbf{R} = -\mathbf{k}$. We also have $\mathbf{J}^{\mathbf{R}} = \mathbf{0}$. Consequently, from (5.21), $\mathbf{L}_1^\top$ becomes

$$\mathbf{L}_1^\top = -\mathbf{k}^\top\mathbf{J}^{\mathbf{V}}\mathbf{L}_{\mathbf{x}} + \mathbf{L}_3^\top \tag{5.23}$$

while $\mathbf{L}_3^\top$ writes $\begin{pmatrix} 0 & 0 & 0 & -2\mathbf{V}^\top\mathbf{j} & -2\mathbf{V}^\top\mathbf{i} & 0 \end{pmatrix}$. Finally, using explicitly $\mathbf{V}$, $\mathbf{J}^{\mathbf{V}}$ and $\mathbf{L}_{\mathbf{x}}$, we simply obtain

$$\mathbf{L}_1^\top = \frac{1}{\| \tilde{x} \|}\left( \frac{x}{\bar{Z}} \; \frac{y}{\bar{Z}} \; -\frac{x^2+y^2}{\bar{Z}} \; y \; -x \; 0 \right) \tag{5.24}$$

where $\bar{Z} = Z^*\| \tilde{\mathbf{x}} \|^2$.

As it can be seen, even if the computation of the vectors $\mathbf{L}_1$ and $\mathbf{L}_2$ to derive the interaction matrix is not straightforward, their final expression is very simple and easy to compute on-line.

## 5.3 Visual Servoing Control Law

The interaction matrix associated to the luminance being known, the control law can be derived. Usually it is based on a desired behavior for the error signal $\mathbf{e}$. More often, an exponential decoupled decrease of this signal is required, that is $\dot{\mathbf{e}} = -\lambda\mathbf{e}$ where $\lambda$ is a positive scalar. Therefore, expressing the temporal derivative of $\mathbf{e}$, we have

$$\dot{\mathbf{e}} = \mathbf{L}_{\mathbf{s}}\mathbf{v} = -\lambda\mathbf{e} \tag{5.25}$$

leading to the classical control law given in (5.2) when considering that only an approximation or an estimation of the interaction matrix is available.

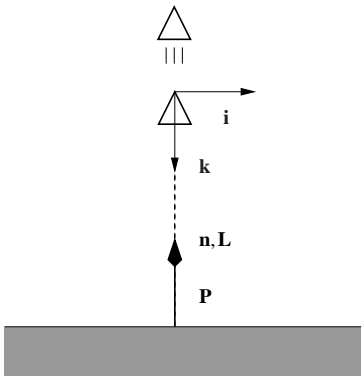


**Fig. 5.2** Light source mounted on the camera for a planar object when the camera and the object planes are parallel.

However, we think that presenting the design of a control law from an optimization problem, as proposed in [16], can lead to more powerful control laws.

### *5.3.1 Visual Servoing as an Optimization Problem*

In that case, the cost function that we have to minimize with respect to the camera current pose writes as follows

$$C(\mathbf{r}) = \frac{1}{2} \parallel \mathbf{e} \parallel^2 \tag{5.26}$$

where $\mathbf{e} = \mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)$.

Nevertheless, regardless of the complexity of the shape of (5.26), since to evaluate (5.26) at a given pose a motion has to be performed, this problem becomes more complex than a classical optimization one if we want to ensure a suitable camera trajectory. Therefore, powerful approaches based on backstepping cannot be used. Indeed, in practice, only differential approaches can be employed to solve this particular optimization problem. In that case, a step of the minimization scheme can be written as follows

$$\mathbf{r}_{k+1} = \mathbf{r}_k \oplus t_k \mathbf{d}(\mathbf{r}_k) \tag{5.27}$$

where "$\oplus$" denotes the operator that combines two consecutive frame transformations; $\mathbf{r}_k$ is the current pose, $t_k$ is a positive scalar (the descent step) and $\mathbf{d}(\mathbf{r}_k)$ a descent direction ensuring that (5.26) decreases if

$$\mathbf{d}(\mathbf{r}_k)^\top \nabla C(\mathbf{r}_k) < 0. \tag{5.28}$$

Consequently, the following velocity control law can be easily derived considering that $t_k$ is small enough

$$\mathbf{v} = \lambda_k \mathbf{d}(\mathbf{r}_k) \tag{5.29}$$

where $\lambda_k$ is a scalar that depends on $t_k$ and on the sampling rate. However, here again, since (5.26) cannot be simply evaluated or estimated, line search algorithms cannot be used and this value is often chosen as a constant one. In the remainder of this chapter we will omit the subscript $k$ for the sake of clarity.

Several descent directions can be used, nevertheless they lead to the following generalized expression of (5.2) (see [6] for more details)

$$\mathbf{v} = -\lambda \widehat{\mathbf{N}_\mathbf{s}} \mathbf{e} \tag{5.30}$$

where:

- $\mathbf{N_s} = \mathbf{L_s}^\top$ for a steepest descent (gradient) method. For instance, this approach has been used in [12];
- $\mathbf{N_s} = \mathbf{L_s}^+$ for a Gauss–Newton (GN) method. It is the control law usually used. Note also that the case where $\mathbf{N_s} = \mathbf{L_{s^*}}^+$ is also very used in practice [10];
- $\mathbf{N_s} = \left(\mathbf{H} + \mu \text{diag}(\mathbf{H})\right)^{-1} \mathbf{L_s}^\top$ for a Levenberg–Marquardt method. $\mathbf{H} = \mathbf{L_s}^\top \mathbf{L_s}$ is an approximation of the Hessian matrix of the cost function (see Section 5.3.2). The parameter $\mu$ makes possible to switch from a steepest descent like approach[2] to a GN one thanks to the observation of (5.26) during the minimization process;
- $\mathbf{N_s} = (\mathbf{L_s} + \mathbf{L_{s^*}})^+$ for the efficient second order minimization (ESM) method proposed in [16]. Note that this method takes benefit of knowing the shape of the cost function near the global minimum (through $\mathbf{L_{s^*}}$); it is thus less sensitive to local minima than the above-mentioned methods. Its convergence domain is also larger.

In practice, since the convergence of the control law (5.30) highly depends on the cost function (5.26), we focus in the next section on its shape.

### *5.3.2 Shape of the Cost Function*

In fact, we are interested in the shape of the cost function since we want to minimize it. Therefore, we are interested in studying the Hessian of (5.26). It is given by

$$\nabla^2 C(\mathbf{r}) = \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\right)^\top \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\right) + \sum_{i=1}^{i=\dim \mathbf{s}} \nabla^2 s_i \left(s_i(\mathbf{r}) - s_i(\mathbf{r}^*)\right). \tag{5.31}$$

However, this expression is far too complex to derive some useful results. Thus, we study it around the desired position $\mathbf{r}^*$, leading to

[2] More precisely, each component of the gradient is scaled according to the diagonal of the Hessian, which leads to larger displacements along the direction where the gradient is low.

$$\nabla^2 C(\mathbf{r}^*) = \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\right)^\top \left(\frac{\partial \mathbf{s}}{\partial \mathbf{r}}\right). \tag{5.32}$$

Moreover, since we have $\dot{\mathbf{s}} = \dfrac{\partial \mathbf{s}}{\partial \mathbf{r}} \dot{r} = \mathbf{L}_\mathbf{s} \mathbf{v}$, we are interested in practice in the following matrix

$$\mathbf{H}^* = \mathbf{L}_{\mathbf{s}^*}^\top \mathbf{L}_{\mathbf{s}^*}. \tag{5.33}$$

This matrix allows us to estimate the cost function around $\mathbf{r}^*$. Indeed, a first order Taylor series expansion of the visual features $\mathbf{s}(\mathbf{r})$ around $\mathbf{r}^*$ gives

$$\mathbf{s}(\mathbf{r}) = \mathbf{s}(\mathbf{r}^*) + \mathbf{L}_{\mathbf{s}^*} \Delta \mathbf{r} \tag{5.34}$$

where $\Delta \mathbf{r}$ denotes the relative pose between $\mathbf{r}$ and $\mathbf{r}^*$. Therefore, by plugging (5.34) into (5.26), we obtain an approximation of the cost function in a neighborhood of $\mathbf{r}^*$

$$\widehat{C}(\mathbf{r}) = \frac{1}{2} \Delta \mathbf{r}^\top \mathbf{H}^* \Delta \mathbf{r}. \tag{5.35}$$

Of course, the graal would be that the eigenvalues of $\mathbf{H}^*$ are the most similar as possible since in that case the cost function would be an hypersphere. Indeed, only a global minimum would exist and a simple steepest descent method would ensure to reach this minimum. Unfortunately, when using the luminance as visual feature, the eigenvalues are very different[3]. On the other hand, the eigenvectors of $\mathbf{H}^*$ point out some directions where the cost function decreases slowly when its associated eigenvalue is low or decreases quickly when its associated eigenvalue is high. That means that the cost function (5.26) presents very narrow valleys. More precisely, an eigenvector associated to a small eigenvalue corresponds to a valley where the cost varies slowly. In contrast, the cost function varies strongly along an orthogonal direction. It can be shown that is in a direction near $\nabla C(\mathbf{r})$ [6]. These preferential directions where the cost function is low are easy explained by the fact that it is very difficult to distinguish in an image an $x$ axis translational motion (respectively $y$) from a $y$ axis rotational motion (respectively $x$). The $z$ axis being the camera optical axis.

### *5.3.3 Control Law*

As shown in Section 5.3.1, several control laws can be used to minimize (5.26). We first used the classical control laws based on the GN approach and the ESM approach [16, 25]. Unfortunately, they all failed, either because they diverged or because they led to unsuitable 3D motion. It is well-known in optimization theory

[3] Note that this phenomenon also holds for most of the geometrical visual features usually used in visual servoing since a term related to the depth always occurs in the translational part of the interaction matrix (see (5.7)).

that minimizing a cost function that presents narrow valleys is a complex problem. Therefore, a new control law has to be derived.

We propose the following algorithm to reach its minimum. The camera is first moved to reach the valleys and next along the axes of the valleys towards the desired pose. It can be easily done by using a control law formally equal to the one used in the Levenberg–Marquardt approach (see Section 5.3.1). However, the way to tune the parameter $\mu$ is different. We denote this method in the remainder of the chapter as modified Levenberg–Marquardt (MLM). As stated in the Section 5.3.2, the first step can be easily done by using a gradient approach, that is by choosing a high value for $\mu$ (typically $\mu = 1$). Once the bottom of valleys has been reached (see [6] for more details), the parameters $\mu$ is forced to decrease to turn the behavior of the algorithm to a GN approach. The resulting control law is then given by

$$\mathbf{v} = -\lambda \left(\mathbf{H} + \mu \, \mathrm{diag}(\mathbf{H})\right)^{-1} \mathbf{L_I}^{\top} \mathbf{e} \tag{5.36}$$

where $\mu$ is not a constant value.

## 5.4 Experimental Results

In all the experiments reported here, the camera is mounted on a 6 DOF gantry robot. Control law is computed on a Core 2 Duo 3 GHz PC running Linux. Image are acquired at 66 Hz using an IEEE 1394 camera with a resolution of $320 \times 240$. The size of the vector $\mathbf{s}$ is then 76800. Despite this size, the interaction matrix $\mathbf{L_I}$ involved in (5.36) can be computed at each iteration if needed.

### *5.4.1 Positioning Tasks under Temporal Luminance Constancy*

We assume in this section that the luminance $I(\mathbf{x})$ at a given pixel is constant. To make this assumption as valid as possible, a diffuse lighting as been used so that $I(\mathbf{x})$ can be considered as constant with respect to to the viewing direction. Moreover, the lighting is also motionless with respect to the scene being observed. In this section, we will first compare the GN and MLM methods and then show that the photometric visual servoing is robust.

#### 5.4.1.1 Comparison between the GN and the MLM Methods

The goal of the first experiment is to compare the control laws based on GN and MLM approaches when a planar object is considered (it is a photo). The initial error pose was $\Delta\mathbf{r}_{init}$ = (5 cm, -23 cm, 5 cm, -12.5°, -8.4°, -15.5°). The desired pose was so that the object and charge-coupled device (CCD) planes are parallel at $Z = Z^* = 80$

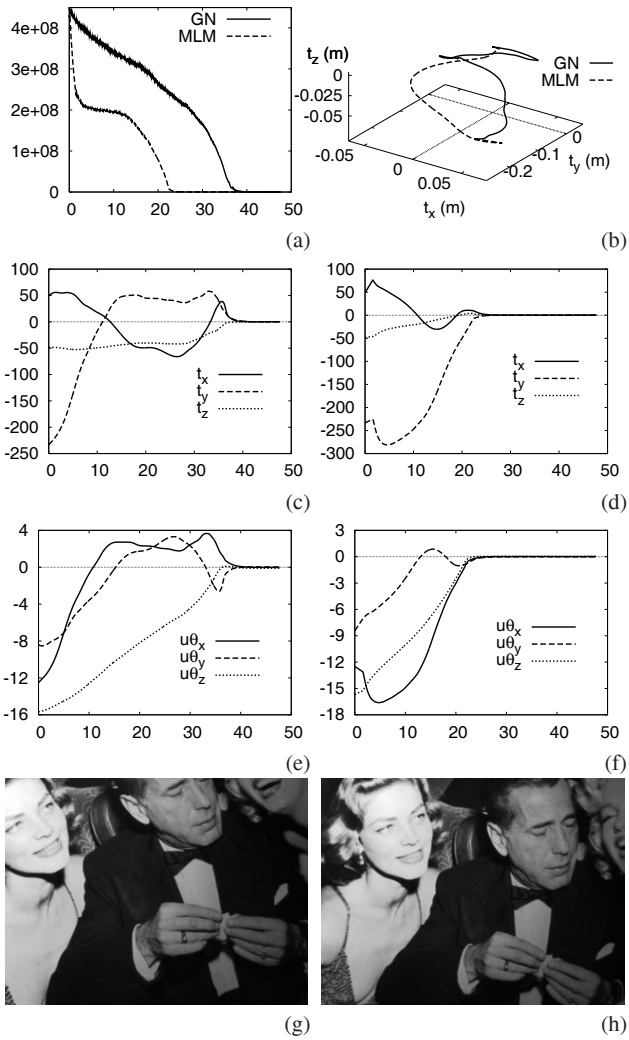


**Fig. 5.3** First experiment, MLM versus GN method ($x$ axis in seconds): (a) comparison of cost functions; (b) comparison of camera trajectories; (c) translation error for the GN method (in mm); (d) translation error for the MLM method (in mm); (e) rotation error for the GN method (in deg); (f) rotation error for the MLM method (in deg); (g) initial image; and (h) final image.

cm. The interaction matrix has been computed at each iteration but assuming that all the depths are constant and equal to $Z^*$, which is of course a coarse approximation.

Fig. 5.3(a) depicts the behavior of cost functions using the GN method or the MLM method while Fig. 5.3(b) depicts the trajectories (expressed in the desired frame) when using either the GN or the MLM method. Fig. 5.3(c) and Fig. 5.3(d) depict respectively the translation errors for the GN and MLM method while Fig.

5.3(e) and Fig. 5.3(f) depict respectively the orientation errors for the GN and MLM method. The initial and final images are reported respectively on Fig. 5.3(g) and Fig. 5.3(h). First, as it can be seen on Fig. 5.3(a), both the control laws converge since the cost functions vanish. However, the time-to-convergence with the GN method is much higher than the one of the MLM method. The trajectory when using the GN method is also shaky compared to the one of the MLM method (Fig. 5.3(b)). Compare also Fig. 5.3(c) with Fig. 5.3(d) and Fig. 5.3(e) with Fig. 5.3(f). The velocity of the camera when using the MLM method is smoother than when using the GN method (Fig. 5.3(d) and Fig. 5.3(c)). This experiment clearly shows that the MLM method outperforms the GN one. Note that in both cases the positioning errors is very low, for the MLM method we obtained $\Delta\mathbf{r}$ = (0.26 mm, 0.30 mm, 0.03 mm, 0.02°, -0.02°, 0.03°). It is very difficult to reach so low positioning errors when using geometric visual features as it is usually done. Indeed, these nice results are obtained because $\mathbf{I} - \mathbf{I}^*$ is very sensitive to the pose $\mathbf{r}$.

#### 5.4.1.2 Influence of the Image Content

The goal of the next experiment is to show that, even if the luminance is used as a visual feature, our approach does not depend too much on the texture of the scene being observed. Fig. 5.4 depicts the behavior of our algorithm for the planar objects respectively shown on Fig. 5.4(a), (c), (e) and (g) (the initial as well as the desired pose is unchanged). As it can be seen, the control law converges in each cases, even in the case of a low textured scene (Fig. 5.4(a) and (c)). Let us point out that similar positioning errors than for the first experiment have been obtained. This result comes from the fact that the shape of the cost functions (5.26) does not depend too much on the image content (as long as the image does not contain periodic patterns or strong changes of the spatial gradient). It always presents narrow valleys that our control law can cope with.

#### 5.4.1.3 Behavior with respect to Partial Occlusions

The third experiment deals with partial occlusions. The desired object pose as well as the initial pose are still unchanged. After having moved the camera to its initial position, an object has been added to the scene, so that the initial image is now the one shown in Fig. 5.5(a) and the desired image is still the one shown in Fig. 5.3(h). Moreover, as seen in Fig. 5.5(b) and Fig. 5.5(c), the object introduced in the scene is also moved by hand during the camera motion which highly increases the occluded surface. Despite that, the control law still converges. Of course, since the desired image is not the true one, the error cannot vanish at the end of the motion (see Fig. 5.5(e)). Nevertheless, the positioning error is not affected by the occlusions (see Fig. 5.5(h) and Fig. 5.5(i)) since the final positioning error is $\Delta\mathbf{r}$ = (-0.1 mm, 2 mm, 0.3 mm, 0.13°, 0.04°, 0.07°). It is very similar with the previous experiments. Compare also Fig. 5.3(d) with Fig. 5.5(h) and Fig. 5.3(f) with Fig. 5.5(i), the

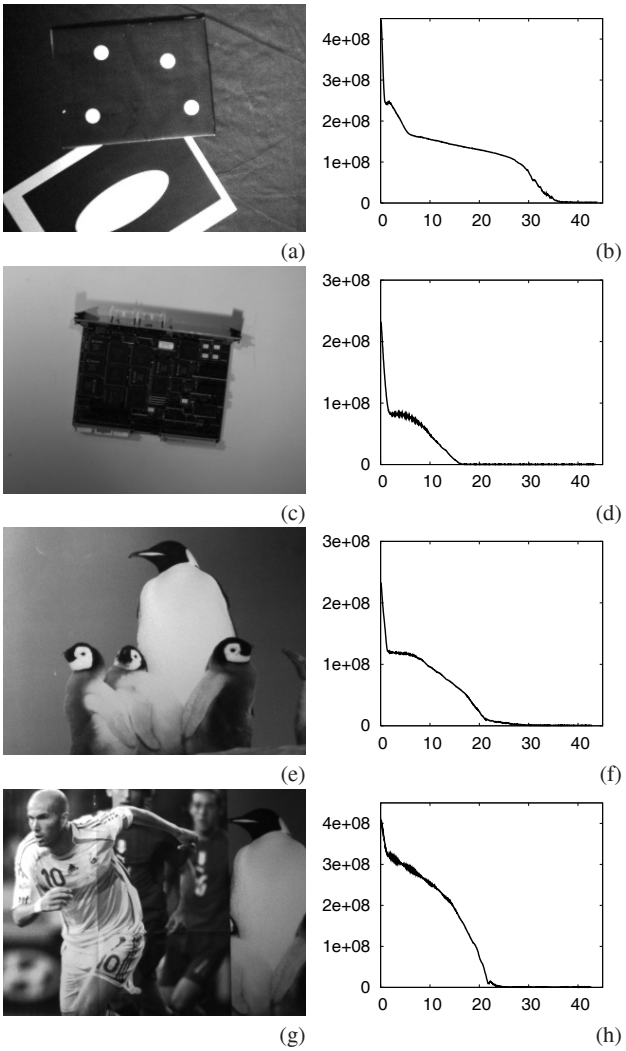


**Fig. 5.4** Second experiment. Same positioning task with respect to various objects. Objects considered (left column) and cost functions (right column) ($x$ axis in seconds).

positioning error, and thus the camera trajectory, are really not affected by the occlusions. This very nice behavior is due to the high redundancy of the visual features we use.

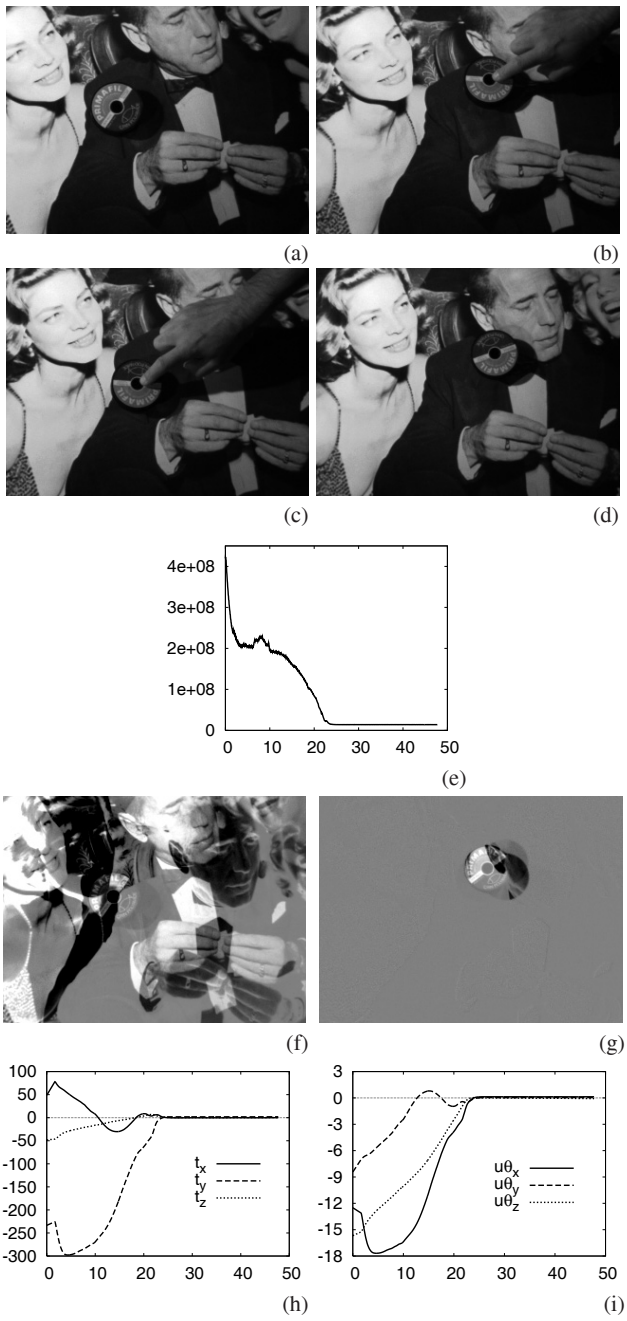


**Fig. 5.5** Third experiment, occlusions ($x$ axis in seconds): (a) initial image; (b) image at t $\approx$ 11 s; (c) image at t $\approx$ 13 s; (d) final image; (e) cost function; (f) $\mathbf{I} - \mathbf{I}^*$ at the initial position; (g) $\mathbf{I} - \mathbf{I}^*$ at the end of the motion; (h) translation errors (in mm); and (i) rotation errors (in deg).

**Fig. 5.6** The nonplanar scene.

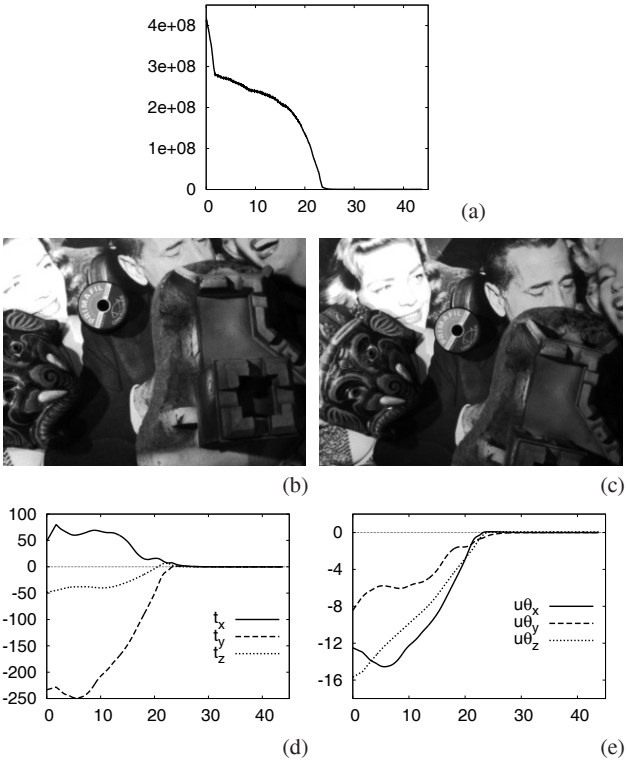


**Fig. 5.7** Fourth experiment, robustness with respect to depths ($x$ axis in seconds): (a) cost function; (b) initial image; (c) final image; (d) translation errors (in mm); and (e) rotation errors (in deg).

#### 5.4.1.4 Robustness to the Depths

The goal of the last experiment is to show the robustness of the control law with respect to the depths. For this purpose, a non planar scene has been used as shown on Fig. 5.6. It shows that large errors in the depth are introduced (the height of the castle tower is around 30 cm). The initial and desired poses are unchanged. Fig. 5.7 depicts this experiment. Here again, the control law still converges (despite the interaction

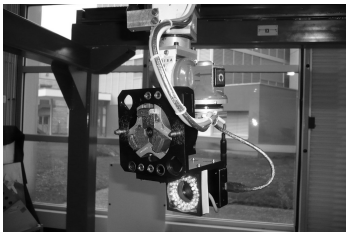

**Fig. 5.8** Camera and light-ring mounted on the robot end-effector.

matrix has been estimated at a constant depth $Z^* = 80$ cm) and the positioning error is still low since we have $\Delta\mathbf{r}$ = (0.2 mm, -0.0 mm, 0.1 mm, -0.01°, 0.00°, 0.06°).

### *5.4.2 Positioning Tasks under Complex Illumination*

In this section we consider the more complex case when the temporal luminance constancy can no more be assumed. Indeed, the scene is no more illuminated by a diffuse lighting since a light-ring is located around the camera lens (see Fig. 5.8). Therefore the light direction is aligned with the camera optical axis as described on Fig. 5.2. This is the unique light in the scene. Note that, obviously, its direction is no more constant with respect to the scene as in Section 5.4.1. The initial positioning error and the desired pose are still unchanged (but with $Z^*$= 70 cm). The interaction matrix has been estimated at the desired position using (5.24) to compute $\mathbf{L}_1^\top$ while $\mathbf{L}_2^\top = 0$ (see the very end of Section 5.2). For all the experiments using the complete interaction matrix we used $k = 100$ and $K_s = 200$ (see (5.19)).

As it can be seen on Fig. 5.9(f), the specularities are very important and consequently their motions in the image are important (for example the specularity can be seen at the bottom of the image in the first image whereas it has moved to the middle at the end of the positioning task). It also almost saturates the image meaning that few information are available around the specularity. The behavior of the control law is better when the complete illumination model is considered since the convergence is faster (see Fig. 5.9(a)). It is also confirmed by observing the positioning errors (compare Fig. 5.9(b) with Fig. 5.9(c) and Fig. 5.9(d) with Fig. 5.9(e)).

Note that tracking tasks and other positioning tasks (when the lighting is not mounted on the camera) have been considered in [6]. These results show, here again, the benefit of using a complete illumination model instead of using the classical temporal luminance constancy.

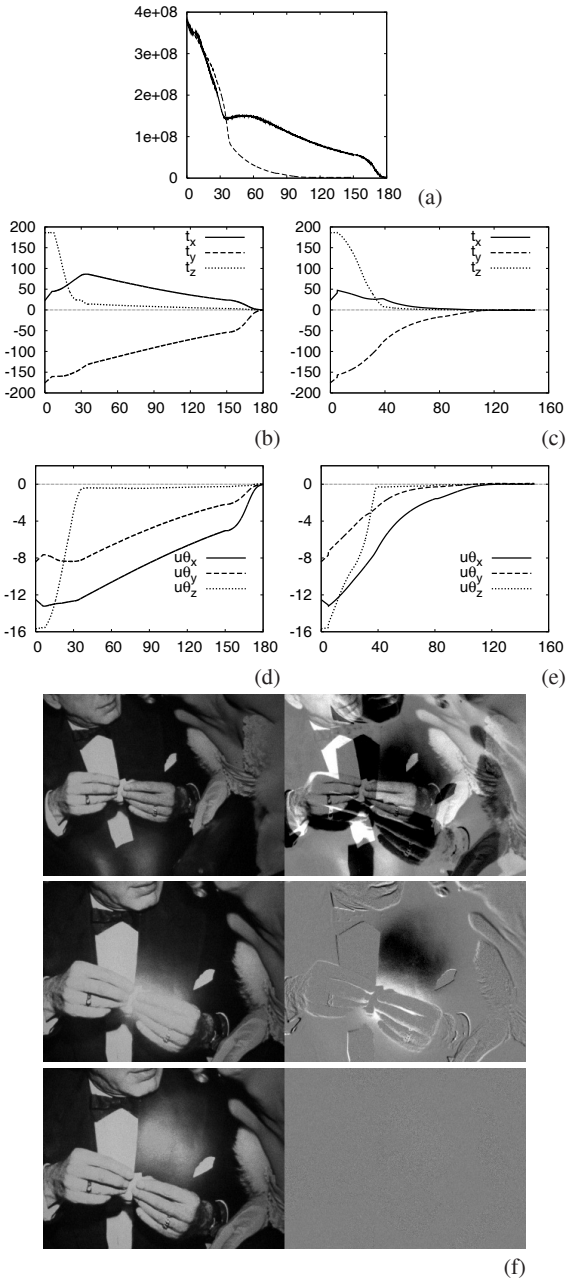


**Fig. 5.9** Positioning task with the light source mounted on the camera: (a) cost function assuming a temporal luminance constancy model (solid line) and using an illumination model (dashed line); (b) translation error assuming a temporal luminance constancy model (in mm); (c) translation error using an illumination model (in mm); (d) rotation error assuming a temporal luminance constancy model (in deg); (e) rotation error using an illumination model (in deg); and (f) images acquired during the positioning task (left) and $\mathbf{I} - \mathbf{I}^*$ (right).

## 5.5 Conclusion and Future Works

We have shown in this chapter the benefit of using the photometric visual servoing. This new visual servoing scheme avoid complex image processing, only remains the image spatial gradient to compute. It also avoid a learning step required with previous approaches based on the use of the image intensity as visual features. This new visual servoing has also other important advantages. Concerning positioning tasks, the positioning errors are always very low. Moreover, this approach is not sensitive to partial occlusions and to coarse approximations of the depths required to compute the interaction matrix. Let us point out that the behavior of the robot is not disturbed by complex illumination changes since the interaction matrix has been derived from a suitable illumination model.

Future work will concern the case when the intensity of the lighting source may vary during the servoing.

**Acknowledgements** The authors wish to thank Francois Chaumette and Seth Hutchinson for their constructive comments.

## References

[1] Abdul Hafez A, Achar S, Jawahar C (2008) Visual servoing based on gaussian mixture models. In: IEEE Int. Conf. on Robotics and Automation, ICRA'08, Pasadena, California, pp 3225–3230

[2] Benhimane S, Malis E (2007) Homography-based 2d visual tracking and servoing. Int Journal of Robotics Research 26(7):661–676

[3] Blinn J (1977) Models of light reflection for computer synthesized pictures. In: ACM Conf. on Computer graphics and interactive techniques, SIGGRAPH'77, San Jose, California, pp 192–198, DOI http://doi.acm.org/10.1145/563858.563893

[4] Chaumette F, Hutchinson S (2008) Visual servoing and visual tracking. In: Siciliano B, Khatib O (eds) Handbook of Robotics, Springer, chap 24, pp 563–583

[5] Collewet C, Marchand E (2008) Modeling complex luminance variations for target tracking. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition, CVPR'08, Anchorage, Alaska, pp 1–7

[6] Collewet C, Marchand E (2008) Photometric visual servoing. Tech. Rep. No. 6631, INRIA

[7] Comport A, Marchand E, Chaumette F (2006) Statistically robust 2D visual servoing. IEEE Trans on Robotics 22(2):415–421, DOI http://dx.doi.org/10.1109/TRO.2006.870666

[8] Crétual A, Chaumette F (2001) Visual servoing based on image motion. Int Journal of Robotics Research 20(11):857–877

[9] Deguchi K (2000) A direct interpretation of dynamic images with camera and object motions for vision guided robot control. Int Journal of Computer Vision 37(1):7–20
[10] Espiau B, Chaumette F, Rives P (1992) A new approach to visual servoing in robotics. IEEE Trans on Robotics and Automation 8(3):313–326
[11] Feddema J, Lee C, Mitchell O (1989) Automatic selection of image features for visual servoing of a robot manipulator. In: IEEE Int. Conf. on Robotics and Automation, ICRA'89, Scottsdale, Arizona, vol 2, pp 832–837
[12] Hashimoto K, Kimura H (1993) LQ optimal and non-linear approaches to visual servoing. In: Hashimoto K (ed) Visual Servoing, vol 7, World Scientific Series in Robotics and Automated Systems, Singapour, pp 165–198
[13] Horn B, Schunck B (1981) Determining optical flow. Artificial Intelligence 17(1-3):185–203
[14] Janabi-Sharifi F, Wilson W (1997) Automatic selection of image features for visual servoing. IEEE Trans on Robotics and Automation 13(6):890–903
[15] Kallem V, Dewan M, Swensen J, Hager G, Cowan N (2007) Kernel-based visual servoing. In: IEEE/RSJ Int. Conf. on Intelligent Robots and System, IROS'07, San Diego, USA
[16] Malis E (2004) Improving vision-based control using efficient second-order minimization techniques. In: IEEE Int. Conf. on Robotics and Automation, ICRA'04, New Orleans, vol 2, pp 1843–1848
[17] Marchand E, Chaumette F (2005) Feature tracking for visual servoing purposes. Robotics and Autonomous Systems 52(1):53–70, DOI http://dx.doi.org/10.1016/j.robot.2005.03.009, special issue on "Advances in Robot Vision", D. Kragic, H. Christensen (Eds.)
[18] Nayar S, Nene S, Murase H (1996) Subspace methods for robot vision. IEEE Trans on Robotics 12(5):750–758
[19] Papanikolopoulos N (1995) Selection of features and evaluation of visual measurements during robotic visual servoing tasks. Journal of Intelligent and Robotic Systems 13:279–304
[20] Phong B (1975) Illumination for computer generated pictures. Communication of the ACM 18(6):311–317
[21] Questa P, Grossmann E, Sandini G (1995) Camera self orientation and docking maneuver using normal flow. In: SPIE AeroSense'95, Orlando, Florida, USA, vol 2488, pp 274–283
[22] Reichmann J (1973) Determination of absorption and scattering coefficients for non homogeneous media. Applied Optics 12:1811–1815
[23] Santos-Victor J, Sandini G (1997) Visual behaviors for docking. Computer Vision and Image Understanding 67(3):223–238
[24] Sundareswaran V, Bouthemy P, Chaumette F (1996) Exploiting image motion for active vision in a visual servoing framework. Int Journal of Robotics Research 15(6):629–645
[25] Tahri O, Mezouar Y (2008) On the efficient second order minimization and image-based visual servoing. In: IEEE Int. Conf. on Robotics and Automation, Pasadena, California, pp 3213–3218